WEST

Help Logout Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

---

Search Results -

Term	Documents
(19 AND 1).USPT.	10
(L19 AND L1).USPT.	10

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear

Search History

DATE: Monday, September 30, 2002 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
DB=USPT; PLUR=YES; OP=ADJ			
L34	l19 and l1	10	L34
L33	l1 same (lock or locking) same reserve same command	3	L33
L32	l1 same reserve same release same command	54	L32
L31	L30 and l1	1	L31
L30	L29 and l28	4	L30
L29	release same command same (unlock or unlocking)	142	L29
L28	(lock or locking) same reserve same command	37	L28
L27	l2 same parity same (update or updating)	130	L27
L26	l1 and l24	9	L26
L25	(l1 or disk or storage) and l24	149	L25
L24	(lock or locking) same (update or updating)same (prevent or preventing) same (access or accessing)	159	L24
L23	l21 same l3	35	L23
L22	L21 and l3	74	L22
L21	(update or updating) same parity	771	L21
L20	l2 and l14 and l17	13	L20
L19	l14 and l17	13	L19
L18	l17 and l16	11	L18
L17	(lock or locking) adj4 parity	33	L17
L16	L14 and l13	141	L16
L15	(exclusive adj ('or' or 'oring'))	22903	L15
L14	(update or updating)adj4 parity	265	L14
L13	l12 same parity	1533	L13
L12	(exclusive adj ('or' or 'oring'))	22903	L12
L11	l10 and l3	118	L11
L10	(update or updating or modify or modifying) same parity	1219	L10
L9	((lock or locking) adj4 parity)and l2	15	L9
L8	L4	37	L8
L7	L6	2	L7
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L6	L5 and l4	3	L6
L5	reserve adj2 command	206	L5
L4	L3 and l2 and l1	42	L4
L3	parity same (lock or locking)	670	L3
L2	RAID	4729	L2
L1	small computer system interface or SCSI	12716	L1

END OF SEARCH HISTORY

> home > about > feedback > logout
US Patent & Trademark Office

Search Results

Search Results for: [parity and RAID and (update or updating)]
Found 79 of 101,942 searched. → Rerun within the Portal

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 79 short listing

Prev Page 1 2 3 4 Next Page

1 Parity logging overcoming the small write problem in redundant disk arrays 100%
Daniel Stodolsky , Garth Gibson , Mark Holland
ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture May 1993
Volume 21 Issue 2

Parity encoded redundant disk arrays provide highly reliable, cost effective secondary storage with high performance for read accesses and large write accesses. Their performance on small writes, however, is much worse than mirrored disks—the traditional, highly reliable, but expensive organization for secondary storage. Unfortunately, small writes are a substantial portion of the I/O workload of many important, demanding applications such as on-line transaction processing. This paper ...

2 Parity logging disk arrays 100%
Daniel Stodolsky , Mark Holland , William V. Courtright , Garth A. Gibson
ACM Transactions on Computer Systems (TOCS) August 1994
Volume 12 Issue 3

Parity-encoded redundant disk arrays provide highly reliable, cost-effective secondary storage with high performance for reads and large writes. Their performance on small writes, however, is

Search Results

Search Results for: [(lock or locking) and parity and RAID and (update or updating)]
Found 25 of 101,942 searched. → Rerun within the Portal

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 20 of 25 **short listing**

Prev Page 1 2 Next Page

1 The architecture of a fault-tolerant cached RAID controller 90%
Jai Menon , Jim Cortney
ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture May 1993
Volume 21 Issue 2

RAID-5 arrays need 4 disk accesses to update a data block—2 to read old data and parity, and 2 to write new data and parity. Schemes previously proposed to improve the update performance of such arrays are the Log-Structured File System [10] and the Floating Parity Approach [6]. Here, we consider a third approach, called Fast Write, which eliminates disk time from the host response time to a write, by using a Non-Volatile Cache in the disk array controller. We examine three alternativ ...

2 RAID: high-performance, reliable secondary storage 88%
Peter M. Chen , Edward K. Lee , Garth A. Gibson , Randy H. Katz , David A. Patterson
ACM Computing Surveys (CSUR) June 1994
Volume 26 Issue 2

Disk arrays were proposed in the 1980s as a way to use parallelism between multiple disks to improve aggregate I/O performance. Today they appear in the product lines of most major computer manufacturers. This article gives a comprehensive overview of disk

Search Results

Search Results for: [parity <near/4> (lock or locking)]
Found 4 of 101,942 searched. → Rerun within the Portal

Results 1 - 4 of 4 **short listing**

1 A simulation model of GECOS III 99%
Kenneth E. Norland , William C. Bulgren
Proceedings of the 1971 annual conference January 1971

A simulation model for a multiprogramming operating system has been devised and programmed in Simscript. Essential elements of the environment have been included such as job arrival rate, maximum number of jobs, the operating system overhead and peripheral and core allocation. Some allowances are made for time-sharing, as well as remote and normal batch jobs. The model is patterned basically after GECOS III, on the H-600 line computer. The hardware constraints considered when necessary are ...

2 The architecture of a fault-tolerant cached RAID controller 99%
Jai Menon , Jim Cortney
ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture May 1993 Volume 21 Issue 2

RAID-5 arrays need 4 disk accesses to update a data block—2 to read old data and parity, and 2 to write new data and parity. Schemes previously proposed to improve the update performance of such arrays are the Log-Structured File System [10] and the Floating Parity Approach [6]. Here, we consider a third approach, called Fast Write, which eliminates disk time from the host response time to a write, by using a Non-Volatile Cache in the disk array controller. We examine three alternativ ...

3 Striping in disk array RM2 enabling the tolerance of double disk failures 99%

Chan-Ik Park , Tae-Young Choe

Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM) November 1996

There is a growing demand in high reliability beyond what current RAID can provide and there are various levels of user demand for data reliability. An efficient data placement scheme called RM2 has been proposed in [11], which makes a disk array system tolerable against double disk failures. In this paper, we consider how to choose an optimal striping unit for RM2 particularly when no workload information is available except read/write ratio. A disk array simulator for RM2 has been developed fo ...

4 RAID: high-performance, reliable secondary storage 98%

Peter M. Chen , Edward K. Lee , Garth A. Gibson , Randy H. Katz , David A. Patterson

ACM Computing Surveys (CSUR) June 1994

Volume 26 Issue 2

Disk arrays were proposed in the 1980s as a way to use parallelism between multiple disks to improve aggregate I/O performance. Today they appear in the product lines of most major computer manufacturers. This article gives a comprehensive overview of disk arrays and provides a framework in which to organize current and future work. First, the article introduces disk technology and reviews the driving forces that have popularized disk arrays: performance and reliability. It discusses the tw ...

Results 1 - 4 of 4 **short listing**

Your search matched **6** of **798921** documents.
Results are shown **25** to a page, sorted by **publication year** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.
Then click **Search Again.**

RAID and parity and (update or updating)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

---

1 **A multi-tier RAID storage system with RAID1 and RAID5**
Muppalaneni, N.; Gopinath, K.
Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International , 2000
Page(s): 663 -671

[Abstract] [PDF Full-Text (236 KB)] **CNF**

---

2 **Improving RAID performance using a multibuffer technique**
Hua, K.A.; Khanh Vu; Ta-Hsiung Hu
Data Engineering, 1999. Proceedings., 15th International Conference on , 1999
Page(s): 79 -86

[Abstract] [PDF Full-Text (108 KB)] **CNF**

---

3 **Double parity sparing for performance improvement in disk arrays**
Min-Young Lee; Myong-Soon Park
Parallel and Distributed Systems, 1996. Proceedings., 1996 International Conference on , 1996
Page(s): 169 -174

[Abstract] [PDF Full-Text (468 KB)] **CNF**

---

4 **Performance study of RAID-5 disk arrays with data and parity cache**
Mishra, S.K.; Mohapatra, P.
Parallel Processing, 1996. Vol.3. Software., Proceedings of the 1996

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore™
RELEASE 1.4

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links

» Search Result

Welcome to IEEE Xplore
- Home
- What Can I Access?
- Log-out

Tables of Contents
- Journals & Magazines
- Conference Proceedings
- Standards

Search
- By Author
- Basic
- Advanced

Member Services
- Join IEEE
- Establish IEEE Web Account

Print Format

Your search matched **[0]** of **[799001]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

RAID and parity and (lock or locking)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Your search matched **4** of **798921** documents.
Results are shown **25** to a page, sorted by **publication year** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.

Then click **Search Again**.

parity and (lock or locking)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Architectural and multiprocessor design verification of the PowerPC 604 data cache**
Cai, G.Z.N.
Computers and Communications, 1995., Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on , 1995
Page(s): 383 -388

[Abstract] [PDF Full-Text (500 KB)] **CNF**

2 **ECC-on-SIMM test challenges**
Dell, T.J.
Test Conference, 1994. Proceedings., International , 1994
Page(s): 511 -515

[Abstract] [PDF Full-Text (344 KB)] **CNF**

3 **Syndrome-based Viterbi decoder node synchronization and out-of-lock detection**
Moeneclaey, M.; Sanders, P.
Global Telecommunications Conference, 1990, and Exhibition. 'Communications: Connecting the Future', GLOBECOM '90., IEEE , 1990
Page(s): 604 -608 vol.1

[Abstract] [PDF Full-Text (348 KB)] **CNF**

4 **Fault-tolerance in a high-speed 2D convolver/correlator: Starloc**

Napolitano, L.M., Jr.; Andaleon, D.D.; Berry, R.R.; Bryson, P.R.; Klapp, S.R.; Leeper, J.E.; Redinbo, G.R.
Fault-Tolerant Computing, 1989. FTCS-19. Digest of Papers., Nineteenth International Symposium on , 1989
Page(s): 80 -87

[Abstract] [PDF Full-Text (756 KB)] **CNF**

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore™ RELEASE 1.4

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Quick Links » Search Result
Peer Review

Welcome to IEEE Xplore

- Home
- What Can I Access?
- Log-out

Tables of Contents

- Journals & Magazines
- Conference Proceedings
- Standards

Search

- By Author
- Basic
- Advanced

Member Services

- Join IEEE
- Establish IEEE Web Account

Print Format

Your search matched **[0]** of **[799001]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

RAID and parity and (lock or locking) and (update or updating)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Your search matched **[0]** of **[798921]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

RAID and parity and (lock or locking)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Your search matched **6** of **798921** documents.
Results are shown **25** to a page, sorted by **publication year** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text box.
Then click **Search Again**.

RAID and parity and (update or updating)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

---

1 **A multi-tier RAID storage system with RAID1 and RAID5**
Muppalaneni, N.; Gopinath, K.
Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings. 14th International , 2000
Page(s): 663 -671

[Abstract] [PDF Full-Text (236 KB)] **CNF**

---

2 **Improving RAID performance using a multibuffer technique**
Hua, K.A.; Khanh Vu; Ta-Hsiung Hu
Data Engineering, 1999. Proceedings., 15th International Conference on , 1999
Page(s): 79 -86

[Abstract] [PDF Full-Text (108 KB)] **CNF**

---

3 **Double parity sparing for performance improvement in disk arrays**
Min-Young Lee; Myong-Soon Park
Parallel and Distributed Systems, 1996. Proceedings., 1996 International Conference on , 1996
Page(s): 169 -174

[Abstract] [PDF Full-Text (468 KB)] **CNF**

---

4 **Performance study of RAID-5 disk arrays with data and parity cache**
Mishra, S.K.; Mohapatra, P.
Parallel Processing, 1996. Vol.3. Software., Proceedings of the 1996

International Conference on , Volume: 1 , 1996
Page(s): 222 -229 vol.1

[Abstract] [PDF Full-Text (736 KB)] **CNF**

---

5 **Algorithms for software and low-cost hardware RAIDs**
Menon, J.; Riegel, J.; Wyllie, J.
Compcon '95.'Technologies for the Information Superhighway', Digest of Papers. , 1995
Page(s): 411 -419

[Abstract] [PDF Full-Text (648 KB)] **CNF**

---

6 **Dynamic parity stripe reorganizations for RAID5 disk arrays**
Mogi, K.; Kitsuregawa, M.
Parallel and Distributed Information Systems, 1994., Proceedings of the Third International Conference on , 1994
Page(s): 17 -26

[Abstract] [PDF Full-Text (780 KB)] **CNF**

WEST

Generate Collection Print

L23: Entry 28 of 35 File: USPT Oct 28, 1997

DOCUMENT-IDENTIFIER: US 5682396 A
TITLE: Control unit in storage subsystem with improvement of redundant data updating

Detailed Description Text (21):
FIG. 22 shows the construction of the PG management information 2001. An empty pointer 2206 is used for linking empty management information 2202 with each other. An update before segment pointer 2200 indicates a segment 1800 in which the update before content of a record 1502 corresponding to the entry is stored. An update after segment pointer 2201 indicates a segment in which the update after value of a record 1502 corresponding to the entry is stored. In the case where both the update before segment pointer 2200 and the update after segment pointer 2201 take null values, it is meant that the corresponding record 1502 is not stored in the cache 1308. A write after bit 2202 is information indicating that a write after process 1313 for a record 1502 corresponding to the entry should be performed. A load request bit 2203 is information indicating that a record 1502 corresponding to the entry should be loaded into the cache 1308. Since the update before segment pointer 2200, the update after segment pointer 2201, the write after bit 2202 and the load request bit 2203 are provided corresponding to each record 1502, the PG management information 2001 includes each of those data which is (m+n) in number equal to the number of records 1502 included in the corresponding parity group 1600. Lock information 2204 indicates that the records 1502 in the parity group 1600 corresponding to the PG management information 2001 under consideration are being operated. In a write action for the disk array using the data distribution by record, not only a data record 1501 but also all parity records 1501 are updated. Therefore, it is required that a write action for the same parity group 1600 is sequentially performed (or serialized) in accordance with the lock information 2204. Lock wait information 2205 is information indicating that a read/write request from the processor 1300 is in a wait condition. The lock wait information 2205 is provided for ensuring that the write action will be performed sequentially. A parity generation bit 2206 is information indicating that records 1502 necessary for generation of updated values of parity records 1501 belonging to the parity group 1600 corresponding to the PG management information 2001 under consideration are stored in the cache 1308.

Detailed Description Text (113):
From the foregoing, in step 3301 shown in FIG. 33, the update before data record 105 is held in the cache 1308 as it is and the write data (corresponding to the update after data record b 6401) accepted in the segment 1800 indicated by the update after segment pointer 2201 is stored into the segment 1800 (corresponding to the update before data record a 6302) in which the update after data record 106 has been stored. In step 3302, a load request bit 2203 corresponding to the update before parity record 107 which has not been loaded in the cache 1308, is turned on. In step 3303, lock information 2204 and disk unit occupy information 2004 are reset. Thereafter, in step 3304, the control unit 1305 reports the completion of the process to the processor 1300.

L24: Entry 27 of 159 File: USPT Jul 10, 2001

DOCUMENT-IDENTIFIER: US 6260115 B1
TITLE: Sequential detection and prestaging methods for a disk storage subsystem

Detailed Description Text (34):
FIG. 10 is a flow diagram of a locking process applied to each list 100. This process prevents conflicts between two or more threads of execution, for example in a multiprocessing environment, attempting to access the same logical storage device data structure simultaneously. Before making any changes to the list 100 for a given logical storage device, the control unit program must check the status of the lock, as shown by decision block 1000. If the list 100 is locked then the control unit software program must either wait for the list 100 to become unlocked, or abandon the attempt to change the list. When the control unit software program finds the list 100 unlocked, it locks the list 100, as shown in block 1002, to prevent interference from another thread of execution of the control unit software program. Next, the locking execution thread may change data within the list 100, as shown in block 1004. When all of the changes are finished, the list 100 is unlocked, as shown in block 1006, so that another thread of execution in the control unit software program can update the list 100.

Detailed Description Text (35):
A variety of locking methods may be used to implement the locking process. For example, a spin-lock word (not shown) could be defined in the control word 104 for each list 100. The spin-lock word allows only one competing thread of execution to access to the list 100, and it prevents that thread of execution from keeping list 100 locked for an indefinite time. Alternatively, a range lock (not shown) may be created to lock and unlock a range of memory addresses. The preferred approach is to use the update in progress bit 106 in the control word 104 to indicate whether the list 100 is currently locked, as shown in FIG. 1. The update in progress bit 106 approach is simple, requires few processor cycles to lock and unlock, and consumes minimal memory.

WEST

End of Result Set

Generate Collection Print

L30: Entry 4 of 4 File: USPT Nov 20, 1984

DOCUMENT-IDENTIFIER: US 4484270 A
TITLE: Centralized hardware control of multisystem access to shared and non-shared subsystems

Detailed Description Text (469):
The SAU Reserve command of FIG. 34 locks the SAU onto the SSP interface that received the command. Other SSPs are denied access to the SAU until the reserving SSP releases the interface or until an SAU Reset command is received from any SSP interface. A busy indication is returned for all commands (except SAU RESET) from other SSPs.

Detailed Description Text (471):
The SAU Release command of FIG. 35 unlocks the SAU from an SSP interface. The release command must be received from the same SSP that caused the reserve condition. A reserve condition results from receipt of a Reserve command or the sending of Unit Check Status.

Detailed Description Text (578):
The expected response from an SSP is first a reserve command which locks the SAU onto the SSP interface that received the command and second, a command to load control store. The set of commands accepted by the SAU following its powering on is limited to these two plus two more: a reset command which causes the SAU to reset its registers and tables, and a release command which countermands the reserve command.

Detailed Description Paragraph Table (8):
TABLE 6 ______________________________________ SAU COMMANDS Code Command 0 1 2 3 4 5 6 7 ______________________________________ Add Subsystem 0 0 0 0 0 1 0 1 05.sub.16 Remove Subsystem 0 0 0 1 0 1 0 1 15.sub.16 Write Subsystem IOP Number 0 0 0 0 1 0 0 1 09.sub.16 Read Subsystem Interface Table 0 0 0 0 1 0 1 0 0A.sub.16 Write IOP State 0 0 1 1 0 0 0 1 31.sub.16 Read IOP State 0 0 1 1 0 0 1 0 32.sub.16 Write SSP Number 0 0 1 0 0 0 0 1 21.sub.16 Read SSP Number 0 0 1 0 0 0 1 0 22.sub.16 Write Control Store 1 0 0 0 0 0 0 1 81.sub.16 Read Control Store 1 0 0 0 0 0 1 0 82.sub.16 SAU Reserve 0 0 0 1 0 0 1 1 13.sub.16 SAU Release 0 0 1 0 0 0 1 1 23.sub.16 SAU Reset 1 1 1 1 1 1 1 1 FF.sub.16 Read ID Word 0 0 0 0 0 1 1 1 0 0E.sub.16 Read ID Word 1 0 0 0 1 0 0 1 0 12.sub.16 Read SPI 0 0 1 0 1 0 1 0 2A.sub.16 Read BCTS Interface 0 0 1 1 1 0 1 0 3A.sub.16 Set Test Mode 0 0 0 1 0 1 1 1 17.sub.16 Clear Test Mode 0 0 1 0 0 1 1 1 27.sub.16 Set SAU Lock 0 1 1 0 0 1 1 1 67.sub.16 Clear SAU Lock 1 0 0 0 0 1 1 1 87.sub.16 Write SSP History 1 0 0 1 0 0 0 1 91.sub.16 Read SSP History 1 0 0 1 0 0 1 0 92.sub.16 ______________________________________

Inlers
1/5/2001

WEST

Generate Collection Print

L24: Entry 35 of 159 File: USPT May 8, 2001

DOCUMENT-IDENTIFIER: US 6230190 B1
TITLE: Shared-everything file storage for clustered system

Detailed Description Text (7):
The file sharing protocol includes the Common Internet File System (CIFS) for Microsoft-based systems or the Network File System (NFS) for Unix-based systems. Alternatively, the file sharing protocols may be the Server Message Block (SMB) protocol, which is used over the Internet on top of its TCP/IP protocol or on top of other network protocols such as IPX or NetBEUI. The file sharing protocol supported by the RAID data storage device 106 or 108 provides a locking facility which may be a file locking facility or a byte-range locking facility. The locking facility enhances data integrity for the file sharing environment of FIG. 1. Locking can be used to coordinate concurrent access to a file by multiple applications and users. It can prevent concurrent readers and writers of shared data from reading "stale" data (i.e., data currently in the process of being updated by another application) and/or overwriting each others' updates.

WEST

Generate Collection Print

L11: Entry 10 of 118 File: USPT May 14, 2002

DOCUMENT-IDENTIFIER: US 6389511 B1
TITLE: On-line data verification and repair in redundant storage system

Detailed Description Text (72):
The general method of verifying and repairing data in a parity redundancy group is analogous to the above method. For example, a primary disk adapter can be selected for examination of a redundancy group on a track-by-track basis, e.g., locking an individual track, writing the entire track from each applicable physical storage device to a cache, and then examining all of the data/parity information for that track before examining the next track. As for the above embodiment, of course, the advantages of the invention may be achieved using other design parameters.

Detailed Description Text (84):
There are two possibilities once the existence of a data coherence problem has been identified. One of the data units may not have been updated, even though the parity information corresponding to that update has been made. In the alternative, one of the data units may have been updated, but the parity unit not updated. Put another way, the writing of data in a parity redundancy group involves writing (1) the new data; and (2) updating parity to correspond to the new data. For example, if data on the D1 track is updated during normal operation, this would result in new data being written to the D1 track and also to the P track. If there is a data coherence problem, it arises from writing either D1 or P, but failing to write the other. In this case, D1 has new/updated data and D1-new has the old (not updated) version of D1, or vice versa. (The first corresponds to failure to update parity, while the second corresponds to a failure to update D1 when parity was updated.)

Detailed Description Text (87):
At a step 122a, the D1-new track is examined to determine if it is a viable track (i.e., is internally consistent with respect to its format and other parameters). If D1-new is not viable, then the coherence problem must have arisen from failure to update data on a different track (or the parity track when that data was written) because D1-new is "garbage."

Detailed Description Text (91):
In the alternative, if D1 bears the newer time stamp, the data coherence problem arose from a failure to update parity when D1 was updated. Accordingly, at a step 124, P is determined invalid and the repair process may proceed.

Detailed Description Text (94):
In the embodiment of FIG. 11, it is assumed at step 122c that the correctness of the format in D1-new is sufficient to identify the exact data coherence problem--failure to update D1 although the P unit was updated, or vice versa. If the nature of the formatting is insufficient to draw this conclusion at this point, then other data tracks may still be examined, by continuing at step 123. If more than one possible data coherence problem is identified, a complete log of the possible data coherence problems can be generated for a further examination of what data coherence problem or problems may exist on this track. This situation may arise, for example, in the event that a write to one data track (e.g., D2) but not parity does not result in a format violation when another track (e.g., D1-new) is generated from parity and the one data track (e.g., D2 XOR P). D1-new will have a correct format but the data coherence problem actually exists elsewhere.

Detailed Description Text (95):

Returning to step 122a, if the D1-new format has been determined not to be viable, then the D1-new unit is simply garbage data. The data coherence problem has not yet been identified--the problem is either a failure to update parity or data on a different disk in the redundancy group. Processing continues, therefore, at a step 123.